

IN THE CLAIMS

Please amend the claims as follows.

For the Examiner's convenience, a list of all claims is included below.

1. (Currently amended) A method comprising:
caching a plurality of successive type hierarchy references corresponding to a data object within ~~the~~ a data structure of the data object; and
accessing the cached type hierarchy references at run time to perform type checking of the data object by
determining, for a query type, that is within the portion of the plurality of successive type hierarchy references, if a data object type corresponding to the data object, is the query type, and if not
comparing a first number corresponding to a number of supertypes depth of the data object type within the plurality of successive type hierarchy references to a second number corresponding to a number of supertypes depth of the query type within the plurality of successive type hierarchy references;
determining that the data object type is not the query type if the first number depth of the ~~data object type~~ is not greater than the second number depth of the query type; and
comparing the query type to a corresponding type hierarchy reference and determining that the data object type is the query type if the query type and the corresponding type hierarchy are equal.

2. (Original) The method of claim 1 wherein the plurality of successive type hierarchy references are cached in a data structure of the data object.
3. (Original) The method of claim 2 wherein the data structure is a data structure of an object oriented computer language.
4. (Original) The method of claim 3 wherein the object oriented computer language is selected from the list consisting of JAVA, C++, C#, and CLI.
5. (Original) The method of claim 1 wherein the plurality of successive type hierarchy references comprises three successive type hierarchy references.
6. (Previously presented) The method of claim 1 wherein the plurality of successive type hierarchy references comprises a maximum number of successive type hierarchy references based upon a specific application requiring type checking of the data object.
7. (Original) The method of claim 1 wherein the plurality of successive type hierarchy references comprises a number of successive type hierarchy references, the number of successive type hierarchy references dynamically determined at run time.
8. (Currently amended) The method of claim 1, wherein the plurality of successive type hierarchy references is a portion of a greater plurality of successive type hierarchy references, further comprising:

a) determining that the query type is not within the plurality of successive type hierarchy references;

b) obtaining a highest type hierarchy reference of the plurality of successive type hierarchy references from the cache;

c) accessing a subsequent data object, the subsequent data object referenced by the highest type hierarchy reference, the subsequent data object having a subsequent portion of the greater plurality of successive type hierarchy references corresponding to the subsequent data object; and

d) accessing the subsequent portion of the greater plurality of successive type hierarchy references and determining that the query type is within the subsequent portion of the greater plurality of successive type hierarchy references ;

e) comparing a first number corresponding to a number of supertypes depth of the data object type within the subsequent portion of the greater plurality of successive type hierarchy references to a second number corresponding to a number of supertypes depth of the query type within the plurality of successive type hierarchy references;

~~e) f)~~ determining that the data object type is not the query type if the first number depth of the data object type is not greater than the second number depth of the query type; and

~~e) g)~~ comparing the query type to a corresponding type hierarchy reference and determining that the data object type is the query type if the query type and the corresponding type hierarchy are equal.

9. (Currently amended) A machine-readable medium provides executable instructions which, when executed by a processor, cause the processor to perform a method, the method comprising:

caching a plurality of successive type hierarchy references corresponding to a data object within ~~the~~ a data structure of the data object; and

 accessing the cached type hierarchy references at run time to perform type checking of the data object by

 determining, for a query type, that is within the portion of the plurality of successive type hierarchy references, if a data object type corresponding to the data object, is the query type, and if not

 comparing a first number corresponding to a number of supertypes ~~depth~~ of the data object type within the plurality of successive type hierarchy references to a second number corresponding to a number of supertypes ~~depth~~ of the query type within the plurality of successive type hierarchy references;

 determining that the data object type is not the query type if the first number ~~depth of the data object type~~ is not greater than the second number ~~depth of the query type~~; and

 comparing the query type to a corresponding type hierarchy reference and determining that the data object type is the query type if the query type and the corresponding type hierarchy are equal.

10. (Original) The machine-readable medium of claim 9 wherein the plurality of successive type hierarchy references are cached in a data structure of the data object.

11. (Original) The machine-readable medium of claim 10 wherein the data structure is a data structure of an object oriented computer language.

12. (Original) The machine-readable medium of claim 11 wherein the object oriented computer language is selected from the list consisting of JAVA, C++, C#, and CLI.
13. (Original) The machine-readable medium of claim 9 wherein the plurality of successive type hierarchy references comprises three successive type hierarchy references.
14. (Previously presented) The machine-readable medium of claim 9 wherein the plurality of successive type hierarchy references comprises a maximum number of successive type hierarchy references based upon a specific application requiring type checking of the data object.
15. (Original) The machine-readable medium of claim 9 wherein the plurality of successive type hierarchy references comprises a number of successive type hierarchy references, the number of successive type hierarchy references dynamically determined at run time.
16. (Currently amended) The machine-readable medium of claim 9 wherein the method further comprises:
 - a) determining that the query type is not within the plurality of successive type hierarchy references;
 - b) obtaining a highest type hierarchy reference of the plurality of successive type hierarchy references from the cache;
 - c) accessing a subsequent data object, the subsequent data object referenced by the highest type hierarchy reference, the subsequent data object having a subsequent portion of the greater plurality of successive type hierarchy references corresponding to the subsequent data object; and

d) accessing the subsequent portion of the greater plurality of successive type hierarchy references and determining that the query type is within the subsequent portion of the greater plurality of successive type hierarchy references ;

e) comparing a first number corresponding to a number of supertypes depth of the data object type within the subsequent portion of the greater plurality of successive type hierarchy references to a second number corresponding to a number of supertypes depth of the query type within the plurality of successive type hierarchy references;

~~d) f) determining that the data object type is not the query type if the first number depth of the data object type is not greater than the second number depth of the query type; and~~

~~e) g) comparing the query type to a corresponding type hierarchy reference and determining that the data object type is the query type if the query type and the corresponding type hierarchy are equal.~~

17 – 23 (Canceled)

24. (Currently amended) An apparatus comprising:

a cache memory having stored therein a plurality of successive type hierarchy references corresponding to a data object;

a main memory having stored therein instructions; and

a processor to execute the instructions such that execution of the instructions causes the processor to access the cached type hierarchy references at run time to perform type checking of the data object by

determining, for a query type, that is within the portion of the plurality of successive type hierarchy references, if a data object type corresponding to the data object, is the query type, and if not

comparing a first number corresponding to a number of supertypes ~~depth~~ of the data object type within the plurality of successive type hierarchy references to a second number corresponding to a number of supertypes ~~depth~~ of the query type within the plurality of successive type hierarchy references;

determining that the data object type is not the query type if the first number ~~depth~~ of the ~~data object type~~ is not greater than the second number ~~depth of the query type~~; and

comparing the query type to a corresponding type hierarchy reference and determining that the data object type is the query type if the query type and the corresponding type hierarchy are equal.

25. (Original) The apparatus of claim 24 wherein the data object includes a data structure, the data structure storing the plurality of successive type hierarchy references.

26. (Original) The apparatus of claim 24 wherein the data object includes a data structure, the data structure storing a pointer to a sub-root log, the sub-root log storing the plurality of successive type hierarchy references.

27. (Original) The apparatus of claim 24 wherein the instructions include code of an object oriented computer language.

28. (Previously presented) The apparatus of claim 24 wherein the plurality of successive type hierarchy references comprises a maximum number of successive type hierarchy references based upon a specific application requiring type checking of the data object..

29. (Original) The apparatus of claim 24 wherein the plurality of successive type hierarchy references comprises a number of successive type hierarchy references, the number of successive type hierarchy references dynamically determined at run time.

30. (Currently amended) The apparatus of claim 24 wherein the main memory stores further instructions such that execution of the further instructions causes the processor to:

a) determining that the query type is not within the plurality of successive type hierarchy references;

b) obtaining a highest type hierarchy reference of the plurality of successive type hierarchy references from the cache;

c) accessing a subsequent data object, the subsequent data object referenced by the highest type hierarchy reference, the subsequent data object having a subsequent portion of the greater plurality of successive type hierarchy references corresponding to the subsequent data object; and

d) accessing the subsequent portion of the greater plurality of successive type hierarchy references and determining that the query type is within the subsequent portion of the greater plurality of successive type hierarchy references ;

e) comparing a first number corresponding to a number of supertypes ~~depth~~ of the data object type within the subsequent portion of the greater plurality of successive type hierarchy

references to a second number corresponding to a number of supertypes ~~depth~~ of the query type within the plurality of successive type hierarchy references;

d) f) determining that the data object type is not the query type if the first number ~~depth~~ ~~of the data object type~~ is not greater than the second number ~~depth of the query type~~; and

e) g) comparing the query type to a corresponding type hierarchy reference and determining that the data object type is the query type if the query type and the corresponding type hierarchy are equal.